Fast wavelet transform domain texture synthesis

D.S.Wickramanayake, E.A.Edirisinghe, H.E.Bez
Parallelism, Algorithms and Architectures Research Centre (PÅRC)
Loughborough University, UK

ABSTRACT

Block based texture synthesis algorithms have shown better results than others as they help to preserve the global structure. Previous research has proposed several approaches in the pixel domain, but little effort has been taken in the synthesis of texture in a multiresolution domain. We propose a multiresolution framework in which coefficient-blocks of the spatio-frequency components of the input texture are efficiently *stitched* together to form the corresponding components of the output texture. We propose two algorithms to this effect. In the first, we use a constant block size throughout the algorithm. In the second, we adaptively split blocks so as to use the largest possible block size in order to preserve the global structure, while maintaining the mismatched error of the overlapped boundaries below a certain error tolerance. Special consideration is given to minimization of the computational cost, throughout the algorithm designs. We show that the adaptation of the multiresolution approach results in a fast, cost-effective, flexible texture synthesis algorithm that is capable of being used in modern, bandwidth-adaptive, real-time imaging applications. A collection of regular and stochastic test textures is used to prove the effectiveness of the proposed algorithm.

Key words: Texture synthesis, image processing, multi resolution, wavelets

1 INTRODUCTION

In adding realism to computer graphics applications, mapping natural textures into computer-generated images is vital. Most of the mapping algorithms used today either uses a synthesized textures or directly synthesis of textures on surfaces. Texture synthesis, where a sample is used to generate. A texture synthesis method starts from a sample image and attempts to produce a texture with a visual appearance similar to that sample, by repeated placement of micro patterns of texture elements on a surface so that when perceived by a human observer, it appears to be generated by the same underlying stochastic process. Unfortunately, creating a robust and general texture synthesis algorithm has been proven difficult.

It is expected to be widely used in texture mappings aimed at improving the realism of computer generated images, whereby texture details are added by wrapping a given texture image around the original surface. So far the most common approach to texture synthesis has been to develop a statistical model, which emulates the generative process of the texture which it is intending to mimic.

Textures have been traditionally classified as either regular or stochastic. Almost all the real world textures lie in between these two extremes. Natural examples of such textures include fur of animals, patterns of flowers, bark on a tree etc., whereas fabric patterns, stone patterns on walls are examples of man-made textures.

The problem of synthesizing textures has been studied extensively and numerous approaches have already been proposed. One approach that has been successful in producing good quality textures uses Markov Random Fields [1,2,3]. This approach has proven to synthesize textures, which are good approximations to a broad range of textures. However the main drawback of these algorithms is their computational intensiveness that prevents them being used in real time texture synthesizing applications.

Another common approach is the physical simulation of the texture. In this method texture generation is done by directly simulating the physical generation process of certain textures such as corrosion, weathering etc. Certain patterns such as fur, scale and skin is modelled using reaction diffusion [4] and cellular texturing [5]. Also some weathering and mineral

Visual Communications and Image Processing 2004, edited by Sethuraman Panchanathan, Bhaskaran Vasudev, Proc. of SPIE-IS&T Electronic Imaging, SPIE Vol. 5308

phenomena can be reproduced by detailed simulations of texture [6]. The main disadvantage of these algorithms is that they cannot be applied to general categories.

Other commonly used approach is the statistical feature matching. In this method certain features of the input texture are matched in constructing the resulting texture. These algorithms are more efficient than Markov Random Field algorithms. Heeger and Bergen [9] proposed a method for modelling textures by matching marginal histograms of image pyramids. This algorithm failed to give good results on structured textures. Simoncelli and Portilla [8] were able to improve the synthesis results on structures by using a complicated optimisation procedure. De Bonet [7] synthesizes the textures from a wide variety of input images by shuffling the elements in the Laplacian pyramid representation. Although this method is better than the [9] for structured textures it can produce boundary artefacts in some cases.

The inspiration for our work comes form the recent algorithm proposed by Efros and Freeman [10]. Their approach is simple and works well with most textures. As there are similarities between Efros's & Freeman's [10] approach and ours, we provide an overview of this algorithm in section 2, and use it as a benchmark to test the proposed algorithm.

For clarity of presentation the rest of the paper is divided into further sections as follows. Section 3 presents the proposed multiresolution framework for texture synthesis. Section 4 provides experimental results and a comprehensive analysis of the results. Finally section 5 concludes, with an insight to possible improvements and future variations.

2 PIXEL BASED IMAGE QUILTING

In [10] Efros and Freeman proposed a patch based texture synthesis algorithm in the pixel domain. The algorithm could be summarised as follows:

The output texture is formed by selectively transferring randomly selected blocks of a predefined size from the input texture image. This is done in two steps, satisfying certain pre-defined criteria. Firstly, given that the top left hand corner block of the output image has been appropriately formed, a subset of blocks from which a good candidate for the block to it's right (assuming a raster scanned order) could be found as follows: All possible blocks of the same block size from the input image is matched to the first block (top left hand corner) of the output image, under a certain overlap. The quality of match between the overlapping areas of two blocks is calculated in terms of the Squared Error (SE, i.e. the L2 norm), as follows:

$$SE = \sum_{p \in O} (I_1(p) - I_2(p))^2 \qquad (1)$$

Where $I_x(p)$ is the intensity value of the pixel p, O is the set representing all pixels belonging to the overlap area of overlapping blocks, I_1 and I_2.

The typical overlap used is 1/3 rd the width of the block. The block having the best matching overlap, and all other blocks whose matching error is within 0.1% of that of the best's is selected as the sub-set from which subsequently a block is picked up randomly, to be the final block selected to be used to patch the output image at the location to the right of the top left hand corner block. This process is continued until the whole output image is formed. In selecting non-boundary type blocks (and the last column of blocks), the overlap considered includes both the overlap with the block in front (as discussed above) and block above. The output image formed following the above procedure is then subjected to a second stage in which each overlapping set of blocks is combined together along a line of best fit, i.e. by performing a minimum error boundary cut, rather than the more obvious straight edge cut.

Unfortunately the above algorithm cannot be used for real time texture synthesis, as its efficiency is relatively low. The use of exhaustive searching in choosing the best match causes computational power to be wasted. Due to the use of a random picking technique (described above) in selecting the final block to be patched with the preceding block, often the seam between the two adjacent blocks are quite visible. Even though a minimum error boundary cutting technique is

used to smoothen off these sudden changes in texture, it involves computationally extensive methodologies such as dynamic programming and thus would not be suitable for real time applications.

In order to resolve the problems discussed above, we propose the use of a multiresolution-framework, which is capable of faster texture synthesis.

3 MULTIRESOLUTION IMAGE QUILTING ALGORITHM

The process of texture synthesis can be mathematically represented by equation (2), where *F* is the texture synthesis function, which takes I_{sample} as the input and synthesizes a texture, I_{output} .

$$F(I_{sample}) = I_{output} \qquad (2)$$

The proposed multiresolution approach starts by applying 2D discrete wavelet transform (Haar Transform) to the sample image, I_{sample}. The application of single level 2D wavelet transform will result in decomposing I_{sample} into a set of component images.

$$I_{sample} = f(I_{sa0}, I_{sh0}, I_{sv0}, I_{sd0}) \qquad (3)$$

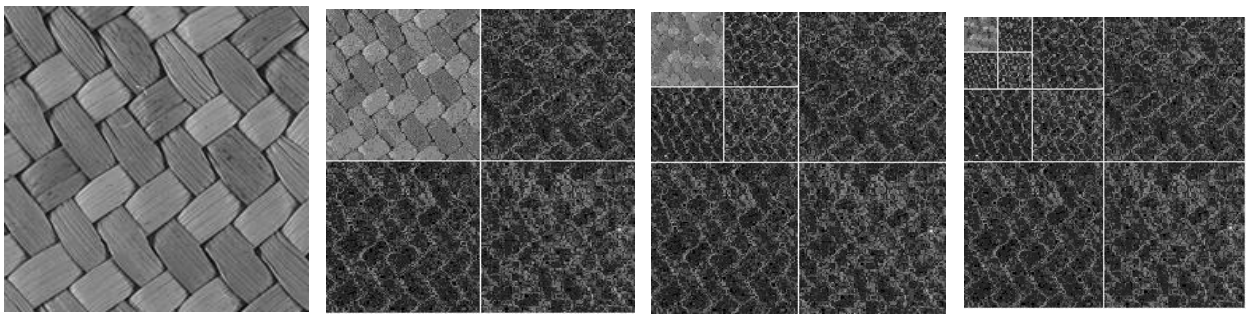
Where $I_{sa0}, I_{sh0}, I_{sv0}, I_{sd0}$ are the component images (coefficient matrices) corresponding respectively to low-resolution approximation, horizontal detail, vertical detail and diagonal detail of the sample input texture. Note that $f = DWT^{-1}$, i.e. f represents the inverse discrete wavelet transform function.

Similarly 2 level and 3 level decompositions are obtained by applying 2D wavelet transform to the corresponding low-resolution component. This could be mathematically presented as follows:

$$I_{sa0} = f(I_{sa1}, I_{sh1}, I_{sv1}, I_{sd1}) \qquad (4)$$

$$I_{sa1} = f(I_{sa2}, I_{sh2}, I_{sv2}, I_{sd2}) \qquad (5)$$

The wavelet decomposition process is visually represented in Figure 1



(a) (b) (c) (d)

Figure 1: Transforming the sample texture into a multiresolution image representation. (a) Sample texture, (b) single level decomposition (c) two level decomposition, (d) three level decomposition

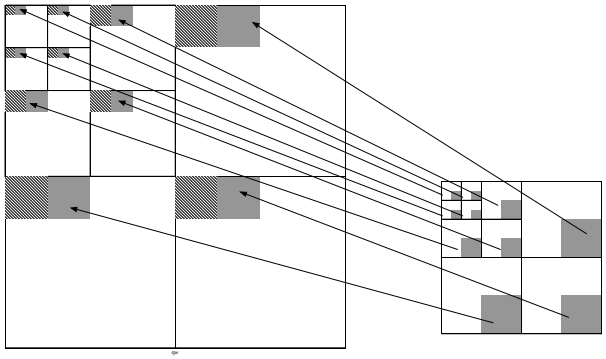
If a wavelet decomposition strategy similar to the above is performed on the output texture, I_{outout} (which is yet to be generated) the following set of equations can be used for its mathematical representation.

$$I_{output} = f(I_{oa0}, I_{oh0}, I_{ov0}, I_{od0}) \qquad (6)$$

$$I_{oa0} = f(I_{oa1}, I_{oh1}, I_{ov1}, I_{od1}) \qquad (7)$$

$$I_{oa1} = f(I_{oa2}, I_{oh2}, I_{ov2}, I_{od2}) \qquad (8)$$

At the start of the proposed method's synthesis stage, we only consider the sample images four components at 3rd level of decomposition, i.e., $I_{sa2}, I_{sh2}, I_{sv2}, I_{sd2}$. (see Figure 1(d)). We first pick a 8×8 block randomly from I_{sa2} and place it in the top left hand corner of the output image, I_{oa2}. Subsequently using above block as the lowest resolution block, all corresponding coefficient blocks of all detailed components are located from the 3-level decomposition of the sample image. They are then transferred to the corresponding locations within appropriate components of the output texture decomposition (see figure 2). Once the first block has been randomly selected and transferred to the output representation, as discussed above, all possible blocks of similar size from the input sample image's I_{sa2} component are picked and matched, for a good overlap with the first block. The matching procedure adapted is different to that used by Efros and Freeman (see section 2) in that it also uses one of the three detail component images at level 3, in deciding the best possible match.



(a) (b)

Figure 2: Construction of the output texture. (a) First random block ( ) and its best match ( ) (second block) placed on top left hand corner of the output texture together with the corresponding coefficient blocks of detail components in all three levels. (b) Selected best match for the first block with its corresponding coefficient blocks from the input texture

The decision to select the horizontal, vertical or diagonal component image is taken by comparing the energy level of these components. This is justified as a higher energy detail component of a particular type; say horizontal detail component, would mean that the horizontal details of the original image would be more significant and visually important than the vertical or diagonal detail. In combining the two overlap errors, i.e. the overlap error between the blocks in the low-resolution component and overlap error between the corresponding blocks in the selected detail component, we use the sum of the squares of the square-errors (Eq. 1) of the two components, as the matching criteria. For example, if diagonal details are prominent the comparison equation is given by the following equation.

$$d(O_1,O_2) = \sum_{p \text{ in } O}\left[\{(I_{oa2}(p) - I_{sa2}(p)\}^2 + \{(I_{od2}(p) - I_{sd2}(p)\}^2\right] \qquad (9)$$

Once the best matching block to the first block of the output image is found and located in the output images I_{oa2} component, the corresponding coefficient blocks from all detail images of the sample texture image are

transferred to the appropriate components of the output texture image. (see figure 2). This process is continued until the whole output texture image is constructed in a three-level decomposed format. Note that when constructing blocks other than those belonging to the first row and first column of blocks, the two overlaps, first corresponding to the overlap with the block in front and second corresponding to the overlap with the block above, needs to be considered. Finally an inverse DWT is performed on the output image formed, to create the output texture image.

In order to maintain the global structure of the overall texture it is important to select the block size as large as possible. This also accounts for the increased efficiency of the algorithm as the choice of blocks available for filling the output texture becomes less, making the process fast. At the same time, selection of large block sizes makes it increasingly difficult to find overlapping areas providing a good match, lowering the quality of the resulting texture. Selection of the optimum size of the block is dependent on the repeating pattern contained in the texture to be synthesized. The use of small block sizes will increase the synthesis time. Thus in an effective implementation of the proposed algorithm we need to have a trade off between the image quality and efficiency in selecting the block size. Optimum trade off can be achieved by the following procedure: We start with a block size of 64× 64 pixels. Subsequently a maximum overlap error (mean-squared error) beyond which the visual quality is poor is defined. Vertical and horizontal overlap errors are measured separately. If the vertical overlap error of a particular block is higher than the relevant threshold the block is split horizontally and vice versa. (see figure 3). Finally, depending on the orientation of the split we search and match, blocks of size 32× 64 or (64× 32) from the input texture, to fill the 64× 64 area under consideration. How ever the process is not repeated iteratively to smaller block sizes due to the need of preserving the global structure.

Figure 3: Changing the block size adaptively. Hatched regions show already synthesized texture. Highlighted regions show overlapped areas (a) If the horizontal overlap error is higher than the threshold, block size is down sized vertically. (b) If vertical overlap error is higher than the threshold; block size is down sized horizontally.

4 EXPERIMENTAL RESULTS AND ANALYSIS

In order to analyse the performance of the proposed algorithm, experiments were performed on a large database of texture images, both regular and stochastic in nature. A typical set of sample images and the output textures obtainable using the proposed texture synthesis algorithm is illustrated in figure 4. The results clearly indicate that the proposed method is capable of providing high quality texture synthesis for a wide variety of textures.

Closer inspection of results in figure 4 and further experiments have revealed that there are certain important factors that are crucial in deciding the quality of the final outcome and the efficiency/computational complexity of the quilting based texture synthesis algorithm. They are:

(1) Level and type of detail used in selecting the matching block.
(2) Size of the unit of construction used.
(3) Overlapping area used in matching.

4.1 Level and type of detail: In contrast to the method proposed by Effros and Freeman, we have adapted a multiresolution matching strategy in selecting the adjacent blocks of the output texture. The use of pixel level detail [10] would not only make the texture synthesis inefficient, but also unsuitable for real time texture synthesis capabilities expected from modern imaging applications.

In the experiments performed we have assumed a typical three-level decomposition resulting in the lowest resolution component image to be 1/64 times the size of the original image. The decisions about matching adjacent blocks are taken only considering the above component and one other detail component (see section-2) from the 3rd level of decomposition. Thus the computational complexity is considerably reduced as compared to the method proposed in [10]. The exclusive use of the low-resolution component is justified due to the fact that it possesses the highest energy of all four components at a given level of decomposition. Further experimental results indicated that the dynamic use of the additional detail component (see section-2) resulted in better matching as compared to using only the lowest resolution component, justifying its use. Further experiments also showed that, using detail images of higher resolution levels, does not significantly improve the quality of the final output texture.

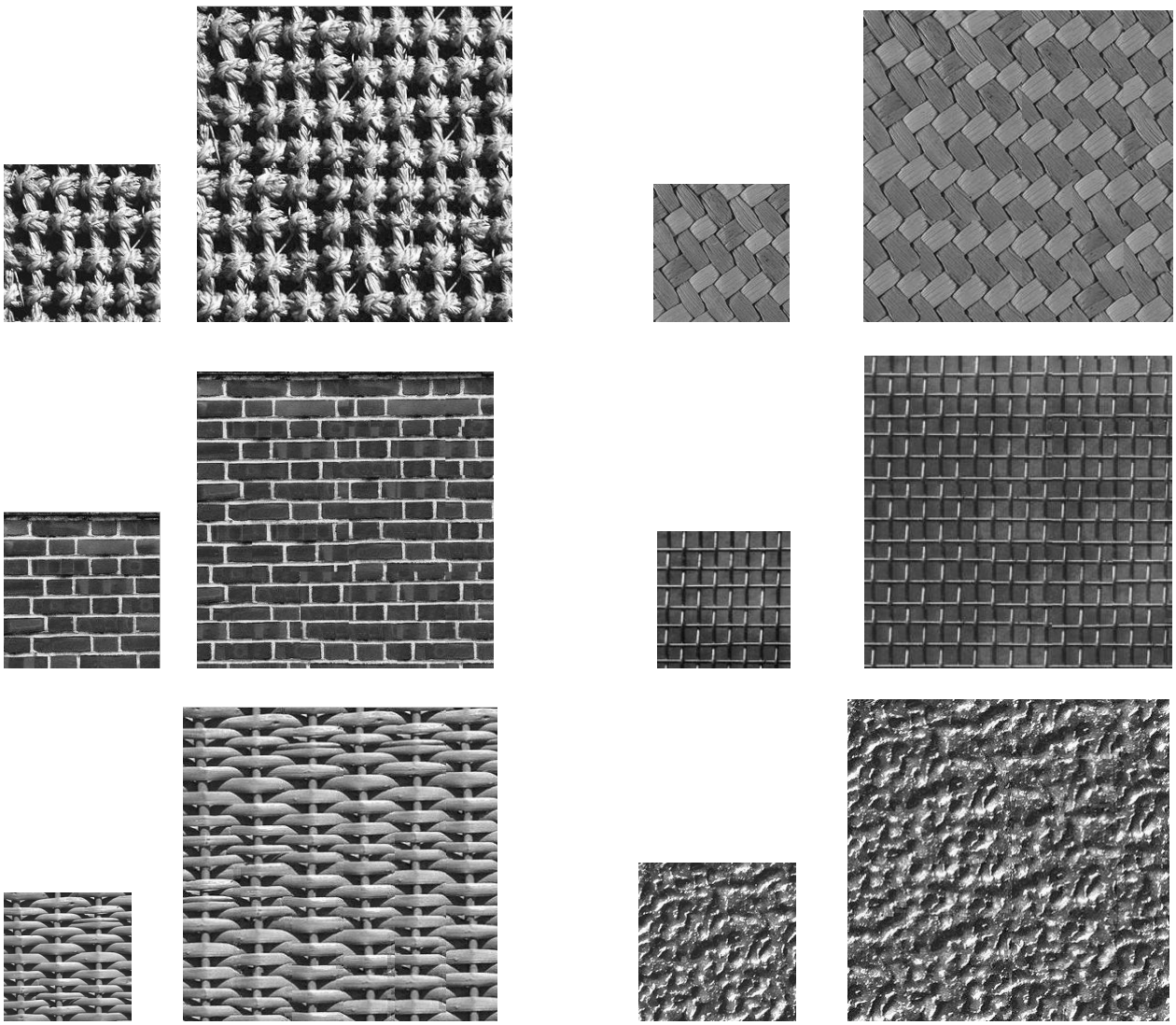


Figure 4: Multiresolution image quilting results. Sample texture and synthesized textures using proposed algorithm

Instead of randomly picking a block from a set of blocks, which match within a certain matching error [10], the proposed texture synthesis method selects the best possible match in terms of the overlap criteria discussed above. The experiments carried out by us revealed that the earlier technique is ineffective if used in conjunction with the proposed multiresolution-based approach. More importantly, this simplification also means that the proposed technique needs less memory capacity to carry out this task.

4.2 Block size: The size of the element used in building the overall texture accounts for the overall quality of the texture and the efficiency of the algorithm as explained above. Selection of the best size of the block is dependent on the repeating pattern contained in the texture to be synthesized. In order to keep the computational complexity of the algorithm low, we have opted to use a fixed block size of 8×8 pixels at the third level of resolution. This is equivalent to using a 64× 64 pixel block at full pixel resolution. In the latter algorithm where we adaptively change the block size, three different block sizes, 8× 8, 4× 8 and 8× 4 are used. Figure 5 shows the results synthesized using adaptive block sizes.

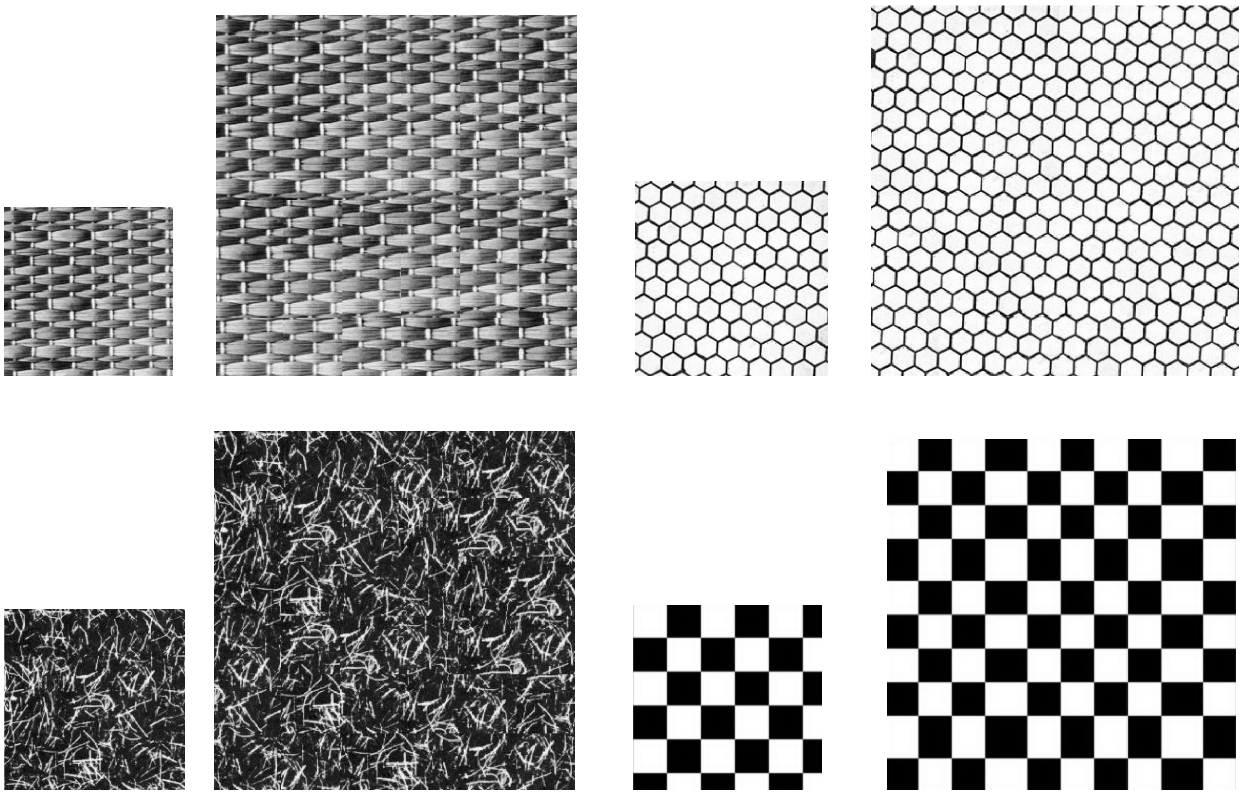


Figure 5: Texture Synthesis results with adaptive block sizes

4.3 Area of overlap: In selecting the matching block, the area of overlap will also account for the quality and the speed of synthesis. Use of less number of overlapping elements (coefficients) results in increased efficiency and more visible artefacts. Increase of Overlapping elements results in better quality with less artefacts and increased synthesis time. However, a too extensive increase in overlapping area will result in noticeable artefacts as it makes it more difficult for the algorithm to make the correct decision on the perceptually best matching block. In order to maintain a compromised situation we have adapted an overlap of a single coefficient row (or column) at the third level of decomposition. This amounts to an overlap of 8 pixel rows (or columns) in the pixel domain.

In general, the use of DWT (Haar transform in our experiments, but could be any) in obtaining the multiresolution decomposition also makes the proposed technique more adaptable to texture synthesis applications which are used in conjunction modern image compression techniques, such as JPEG-2000. In figure 6, we illustrate the possibility of

obtaining the texture synthesis of the output texture in a multiresolution environment. This is of critical importance to modern and future interactive media applications, which effectively deals with progressive/interactive image/video transmission.

Figure 7, compares the performance of the proposed algorithm with that of Effro's and Freeman's. The results illustrate that the perceptual quality of the proposed technique is equivalent to that of [10].

(a) (b) (c) (d) (e)

Figure 6: Input texture and output texture constructed using coefficients at different levels (a) Input texture (I_{sample}). Output texture constructed, (b) using only low-resolution details of the third level (I_{oa2}) (c) using all 3rd level details (I_{oa1}) (d) using 3rd and 2nd level details (I_{oa0}) (e) using all details of all levels (I_{output})

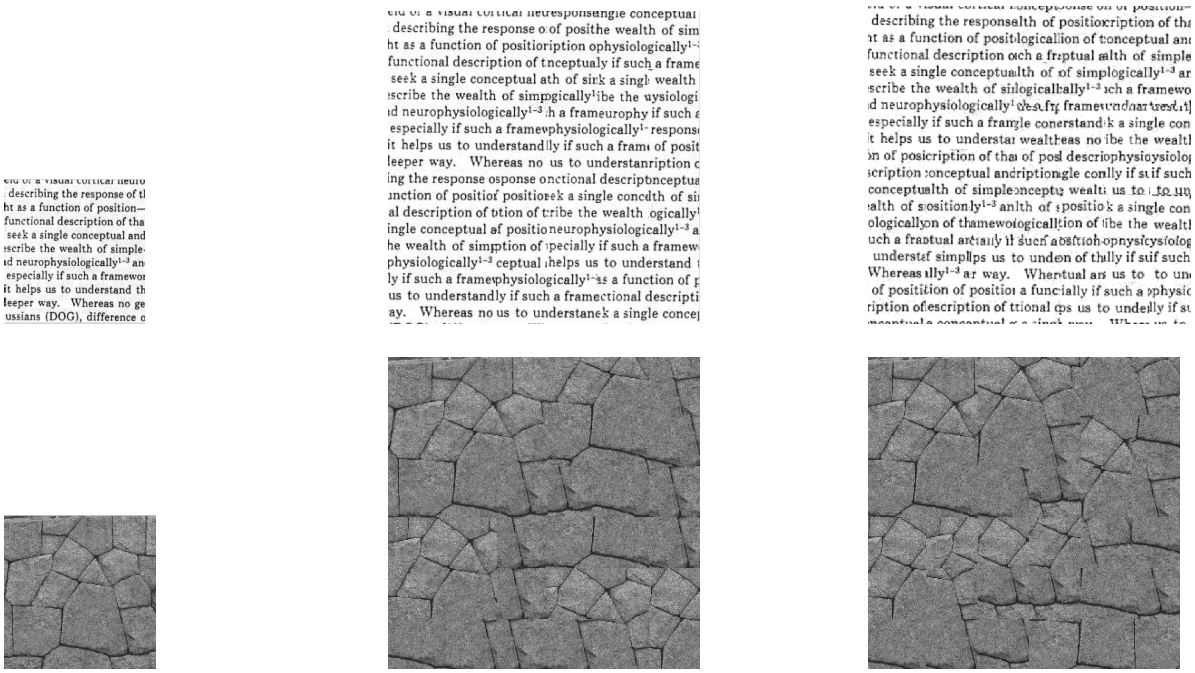


Figure 7: Comparison of performance of the purposed algorithm with Efros and Freeman algorithm. Left: Sample input textures. Middle: Output textures generated using proposed algorithm. Right: Output texture generated using Efros and Freeman algorithm.

5 CONCLUSION AND FUTURE WORK

In this paper we have introduced a novel approach to synthesizing textures under a multi resolution framework. We have provided experimental results and an in-depth analysis, proving that the proposed method works remarkably well, producing equivalent (or better) output texture quality as compared to the method proposed in [10], as a much less computational cost. The multiresolution nature of the proposed framework also makes it easily applicable to modern imaging applications needing progressive transmission capabilities. Figure

At present we are extending the idea further in two directions, namely, increasing the implementation efficiency and preventing seam visibility using adaptive edge cutting techniques. Finally we aim to test the proposed algorithm in mapping real texture onto the surfaces of modelled 3D objects.

REFERENCES

[1] A.Efros and T.Leung. "Texture synthesis by non-parametric sampling". In *International Conference for Computer Vision*, Volume 2, pages 1033-1038, September 1999.

[2] K.Popat and R.Picard. "Novel cluster based probability model for texture synthesis", classification and compression. In *Visual Communication and Image Processing*. Pages 756-768, 1993.

[3] R.Paget and I Longstaff. "Texture synthesis via noncasual nonparametric multi-scale Markov Random Field". *IEEE Transactions on Image Processing*, **7(6):** 925-931, June 1998.

[4] A.Witkin and M.Kass. Reaction-diffusion textures. In Computer *Graphics (SIGGRAPH '91 Proceedings)*, July 1991.

[5] S.P.Worly. "A cellular texture basis function". In H. Rushmeirer, editor, *SIGGRAPH '96 conference proceedings*, Annual conference series, pages 291-294, Aug 1996.

[6] J.Dorsey, E.Edelman, J.Legakis, H.W.Jensen and H.K.Pedersan. "Modelling and rendering of weathered stone". *Proceeding of SIGGRAPH '99*, Pages 225-234, and August 1999.

[7] J.S.De Bonet. "Multiresolution sampling procedure for analysis and synthesis of texture images". In T.Whitted, editor, *SIGGRAPH 97, Conference Proceedings*, Annual conference series, Pages 361-368, August 1997.

[8] E.Simonceli and J.Portilla. "Texture characterization via joint statistics of wavelet coefficient magnitudes". In *Fifth International conference on image processing*, Volume 1, pages 62-66, October 1998.

[9] D.J.Heeger and J.R.Bergen. "Pyramid-Based texture analysis/synthesis". In *SIGGRAPH '95*, pages 229-238, August 1995.

[10] A.Efros and W. T. Freeman, "Image Quilting for Texture Synthesis and Transfer", *Proceedings of SIGGRAPH '01*, pages 341-346, Los Angeles, California, August, 2001

[11] J.R.Bergen and M.S.Landy. "Computational modelling of visual texture segregation", In M.S.Landy and J.A.Movshon, editors, *Computational models of visual perception*, pages 253-271. MIT Press, Cambridge MA, 1991.

[12] C.Chubb and M.S.Landy. "Orthogonal distribution analysis: A new approach to the study of texture perception". M.S.Landy and J.A.Movshon, editors, *Computational models of visual perception*, pages 291-301. MIT Press, Cambridge MA, 1991.